



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/628,599	07/28/2000	Ian R. Finlay	12780-1015	3713

7590

07/31/2006

Sandra M Parker
Attorney at Law
329 La Jolla Aveune
Long Beach, CA 90803

EXAMINER

LY, ANH

ART UNIT	PAPER NUMBER
----------	--------------

2162

DATE MAILED: 07/31/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

DETAILED ACTION

1. This Office Action is response to Applicants' RESPONSE filed on 05/08/2006.
2. Claims 1-22 are pending in this Application.

Response to Arguments

Applicant argued that, "Chow does not teaches "a direct call replacing a lookup function of the run time interpreter."

The applied reference, US Patent No.: 5,875,334 issued to Chow et al. (hereinafter Chow) teaches pre-processing a SQL query statement is processed through a query compiler for generating a access executable plan for a database system such as IBM's DB2; this process requiring a run-time interpreter for looking up local variable; also this executable plan includes a plurality of SQL3 Query control statements, which are parsing into some operation codes into query graph model (QGM) representation of the statements, from which they are then processed to optimized QGM, an access execution plan such as cost-based optimized access plan and they are to be produced operation code or opcode by code generation module (item 118: abstract, col. 1, lines 15-35, col. 6, lines 25-67 and col. 8, lines 8-58; also col. 10, lines 1-40).

Applicant argued that, ""determining from the access plan an executable function associated with a first operation code" does found in Chow reference."

Chow teaches access operations with control statements, optimization code generation and threaded code generated by the compiler and the query extractor for

Art Unit: 2162

determining from access plan associated with control code (col. 14, lines 56-67 and col. 15, lines 1-42; and col. 18, lines 52-67 and col. 19, lines 1-6; fig. 1, parser, the optimized QGM, and threaded code generation are used to produce operation code of SQL query statements to get the optimized access plan or access execution plan : figs. 1 and 2, item 115 & 118, col. 6, lines 25-67 and col. 8, lines 8-58, col. 14, lines 40-67 and col. 15, lines 1-67 and col. 16, lines 1-67).

Applicant argued that, "Chow does not teaches or mention augmenting or replacing opcode."

Chow teaches the extracted SQL statement or control statements is used to produce opcode based on parser, QGM and code generation and replacing this opcode (fig.2 and col. 15, lines 45-67 and col. 17, lines 25-67).

Applicant argued that, "Chow does not teach use of pointer to the opcode."

Chow teaches replacing these opcode by a lookup function to look referencing variables storing in the symbol table as the value of addressing or pointers (fig. 4, col. 24, lines 36-67 and col. 25, lines 1-8).

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in a patent granted on an application for patent by another filed in the United States before the invention thereof by the applicant for patent, or on an international application by another who has fulfilled the requirements of paragraphs (1), (2), and (4) of section 371(c) of this title before the invention thereof by the applicant for patent.

The changes made to 35 U.S.C. 102(e) by the American Inventors Protection Act of 1999 (AIPA) and the Intellectual Property and High Technology Technical Amendments Act of 2002 do not apply when the reference is a U.S. patent resulting directly or indirectly from an international application filed before November 29, 2000. Therefore, the prior art date of the reference is determined under 35 U.S.C. 102(e) prior to the amendment by the AIPA (pre-AIPA 35 U.S.C. 102(e)).

4. Claims 1-8, 10-17 and 19-22 are rejected under 35 U.S.C. 102(e) as being anticipated by US Patent No.: 5,875,334 issued to Chow et al. (hereinafter Chow).

The applied reference has a common assignee with the instant application. Based upon the earlier effective U.S. filing date of the reference, it constitutes prior art under 35 U.S.C. 102(e). This rejection under 35 U.S.C. 102(e) might be overcome either by a showing under 37 CFR 1.132 that any invention disclosed but not claimed in the reference was derived from the inventor of this application and is thus not the invention "by another," or by an appropriate showing under 37 CFR 1.131.

With respect to claim 1, Chow teaches a method for pre-processing an access plan generated for a query in a relational database management system to include said

Art Unit: 2162

access plan including a plurality of operation codes, each of said operation codes being associated with one or more executable functions for performing the query (figs. 1-3, Pre-processing a SQL query statement is processed through a query compiler for generating a access executable plan for a database system such as IBM's DB2; this process requiring a run-time interpreter for looking up local variable; also this executable plan includes a plurality of SQL3 Query control statements, which are parsing into some operation codes into query graph model (QGM) representation of the statements, from which they are then processed to optimized QGM, an access execution plan such as cost-based optimized access plan and they are to be produced operation code or opcode by code generation module (item 118): abstract, col. 1, lines 15-35, col. 6, lines 25-67 and col. 8, lines 8-58; also col. 10, lines 1-40);

determining from the access plan an executable function associated with a first operation code (fig. 1, parser, the optimized QGM, and threaded code generation are used to produce operation code of SQL query statements to get the optimized access plan or access execution plan : figs. 1 and 2, item 115 & 118, col. 6, lines 25-67 and col. 8, lines 8-58, col. 14, lines 40-67 and col. 15, lines 1-67 and col. 16, lines 1-67); and

augmenting said first operation code in the access plan with a pointer to said executable function to provide a direct call mechanism replacing lookup function of a runtime interpreter (figs. 1 & 2, the extracted SQL statement or control statements is used to produce opcode based on parser, QGM and code generation and replacing this opcode (fig.2 and col. 15, lines 45-67 and col. 17, lines 25-67); the executable function in the run-time interpreter with a compiler generated local variables storing in symbol

Art Unit: 2162

table for looking up with referencing local variables or pointers or indexes: fig. 4, col. 24, lines 36-67 and col. 25, lines 1-8; also, col. 15, lines 1-67, col. 17, lines 1-28, col. 20, lines 1-45 and col. 21, lines 5-55).

With respect to claim 2, Chow teaches comprising repeating steps (a) and (b) for remaining operation codes in the access plan (repeating the process with the SQL query statements with loop statement each time the function statement is called: col. 10, lines 12-18 and col. 12, lines 10-22).

With respect to claim 3, Chow teaches wherein said step (b) comprises augmenting said first operation code in the access plan with a pointer to an intermediate function, said intermediate function including a data structure for storing a pointer to said executable function (replacing these opcode by a lookup function to look referencing variables storing in the symbol table as the value of addressing or pointers: fig. 4, col. 24, lines 36-67 and col. 25, lines 1-8).

With respect to claim 4, Chow teaches wherein said data structure includes means for storing information associated with said executable function for said first operation code (fig. 4, symbol table, item 130 & 119, col. 24, lines 36-67).

With respect to claim 5, Chow teaches wherein said step (b) comprises augmenting said first operation code in the access plan with a second pointer to a data structure, said data structure providing means for storing information associated with said first operation code or said executable function (in the symbol table has more than one local variables, each for each SQL query plan, thus there is a second pointer associated with the local variable or operation code: fig. 4, col. 24, lines 36-67).

With respect to claim 6, Chow teaches wherein said step (a) further includes assessing the executable function associated with the first operation code and if applicable, replacing the call to the executable function with a call to a second executable (call statement invoking the SQL statement and replacing the data statement: col. 14, lines 38-67 and col. 15, lines 1-67).

With respect to claim 7, Chow teaches wherein said intermediate function includes processing operations for the first operation code or the executable function associated with the first operation code (col. 23, lines 45-67 and col. 24, lines 1-42).

With respect to claim 8, Chow teaches wherein said processing operations in the intermediate function include gathering statistics on the user of the executable function associated with the operation code (fig. 1, item 301, analyzer generating information for the control flow and scopes and symbol table such as statistical information: col. 18, lines 12-40).

Claim 10 is essentially the same as claim 1 except that it is directed to a computer program product rather than a method, and is rejected for the same reason as applied to the claim 1 hereinabove.

Claim 11 is essentially the same as claim 2 except that it is directed to a computer program product rather than a method, and is rejected for the same reason as applied to the claim 2 hereinabove.

Claim 12 is essentially the same as claim 3 except that it is directed to a computer program product rather than a method, and is rejected for the same reason as applied to the claim 3 hereinabove.

Claim 13 is essentially the same as claim 4 except that it is directed to a computer program product rather than a method, and is rejected for the same reason as applied to the claim 4 hereinabove.

Claim 14 is essentially the same as claim 5 except that it is directed to a computer program product rather than a method, and is rejected for the same reason as applied to the claim 5 hereinabove.

Claim 15 is essentially the same as claim 6 except that it is directed to a computer program product rather than a method, and is rejected for the same reason as applied to the claim 6 hereinabove.

Claim 16 is essentially the same as claim 7 except that it is directed to a computer program product rather than a method, and is rejected for the same reason as applied to the claim 7 hereinabove.

Claim 17 is essentially the same as claim 8 except that it is directed to a computer program product rather than a method, and is rejected for the same reason as applied to the claim 8 hereinabove.

Claim 19 is essentially the same as claim 1 except that it is directed to a system rather than a method, and is rejected for the same reason as applied to the claim 1 hereinabove.

Claim 20 is essentially the same as claim 3 except that it is directed to a computer program product rather than a method, and is rejected for the same reason as applied to the claim 3 hereinabove.

Claim 21 is essentially the same as claim 4 except that it is directed to a computer program product rather than a method, and is rejected for the same reason as applied to the claim 4 hereinabove.

Claim 22 is essentially the same as claim 5 except that it is directed to a computer program product rather than a method, and is rejected for the same reason as applied to the claim 5 hereinabove.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

6. This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

Art Unit: 2162

7. Claims 9 and 18 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent No.: 5,875,334 issued to Chow et al. (hereinafter Chow) in view of US Patent No.: 6,077,312 issued to Bates et al. (hereinafter Bates).

With respect to claim 9, Chow teaches a method as discussed in claim 1.

Chow teaches a SQL query statement is processed through a query compiler for generating a access executable plan for a database system such as IBM's DB2; this process requiring a run-time interpreter and for looking up local variable; also this executable plan includes a plurality of SQL3 Query control statement for parsing into some operation codes from query graph model (QGM) representation of the statement to produce an operation code based on access executable plan, replacing the executable function in the run-time interpreter with a compiler generated local variable for looking up the symbol table with referencing local variables or pointers. Chow does not clearly teach a pause for receiving user input before or after the call to the executable function.

However, Bates teaches halt execution of a computer program when the computer meets a predetermined criteria (abstract).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Chow the teachings of Bates. One having ordinary skill in the art would have found it motivated to utilize the halt execution of a program when it is needed as disclosed (Bates' abstract), into the system of Chow for the purpose of including a pause for executing a program to replace

Art Unit: 2162

opcode, thereby, helping to locate and identify errors in a program under development (Bates' col. 1, lines 10-20).

Claim 18 is essentially the same as claim 9 except that it is directed to a computer program product rather than a method, and is rejected for the same reason as applied to the claim 9 hereinabove.

Conclusion


8. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire **THREE MONTHS** from the mailing date of this action. In the event a first reply is filed within **TWO MONTHS** of the mailing date of this final action and the advisory action is not mailed until after the end of the **THREE-MONTH** shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than **SIX MONTHS** from the mailing date of this final action.

Contact Information

9. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Anh Ly whose telephone number is (571) 272-4039 or via E-Mail: ANH.LY@USPTO.GOV (**Written Authorization being given by Applicant (MPEP 502.03 [R-2])) or fax to (571) 273-4039 (Examiner's personal Fax No.)**). The examiner can normally be reached on TUESDAY – THURSDAY from 8:30 AM – 3:30 PM. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Breene, can be reached on (571) 272-4107 or **Primary Examiner: Jean Corrielus (571) 272-4032**.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). Any response to this action should be mailed to: Commissioner of Patents and Trademarks, Washington, D.C. 20231, or faxed to: **Central Fax Center: (571) 273-8300**

ANH LY 
JUL. 24th, 2006



JEAN M. CORRIELUS
PRIMARY EXAMINER

Exhibit A

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**PATENT**

Applicant: Finlay et al. Examiner: Anh Ly
Serial No. 09/628,599 Group Art Unit: 2172
Filed: July 28, 2000 Docket No. CA990018US1
Title: DIRECT CALL THREADED CODE

5 CERTIFICATE UNDER 37 C.F.R. § 1.8

10 I hereby certify that this correspondence and identified enclosures are being transmitted by facsimile, to the central FAX # 571/273-8300 and Examiner's fax #571/273-4039, under 37 C.F.R. § 1.8, and is addressed to the Commissioner for Patents, BOX: Amendment, on July 23, 2006.


Sandra Parker

PROPOSED AMENDMENT

15 Mail Stop: Amendment
Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

20 Sir:

Responsive to the telephonic conference with Primary Examiner Jean Corrielus, of July 20 and 21, 2006, please kindly enter the following Amendment in order to expedite prosecution.

IN THE CLAIMS

Please amend claims 3-9, 12-13, 15-18 and 20-22, shown below in the set of all existing
25 claims, as follows. No additional fees are required because the Amendment still has 22 claims, as
filed.

1 1. **(Previously Amended)** A method for pre-processing an access plan generated for
2 a query in a relational database management system to include a direct call mechanism replacing
3 a lookup function of a run-time interpreter, said access plan including a plurality of operation
4 codes, each of said operation codes being associated with one or more executable functions for
5 performing the query, said method comprising the steps of:

6 (a) determining from the access plan an executable function associated with a first
7 operation code; and

8 (b) augmenting said first operation code in the access plan with a pointer to said
9 executable function to provide a direct call mechanism replacing a lookup function of a run-time
10 interpreter.

1 2. **(Original)** The method as claimed in claim 1, further comprising repeating steps (a) and
2 (b) for the remaining operation codes in the access plan.

1 3. **(Currently Amended)** The method as claimed in claim 1, wherein said step (b)
2 comprises:

3 augmenting said first operation code in the access plan with a replacement pointer to an
4 intermediate function, ~~said intermediate function including~~ having a data structure, and
5 ~~for storing a the~~ pointer to said executable function in the data structure.

1 4. **(Currently Amended)** The method as claimed in claim 3, wherein said data structure
2 ~~includes~~ comprises means for storing information associated with said executable function or
3 said first operation code.

1 5. (Currently Amended) The method as claimed in claim 1, wherein said step (b)
2 comprises augmenting said first operation code in the access plan with another ~~a second~~ pointer
3 to a data structure, said data structure providing means for storing information associated with
4 said first operation code or said executable function.

1 6. (Currently Amended) The method as claimed in claim 1, ~~wherein said step (a) further~~
2 comprising a step of assessing the executable function associated with the first operation code ~~to~~
3 determine whether a more specific executable function exists and, if found ~~applicable~~, replacing
4 the ~~call to the~~ executable function with the ~~a call to a second~~ more specific executable function.

1 7. (Currently Amended) The method as claimed in claim 3, wherein said intermediate
2 function comprises ~~includes~~ processing operations for the first operation code or the executable
3 function associated with the first operation code.

1 8. (Currently Amended) The method as claimed in claim 7, wherein said processing
2 operations in the intermediate function comprise ~~include~~ gathering statistics on the use of the
3 executable function associated with the first operation code.

1 9. (Currently Amended) The method as claimed in claim 7, wherein said processing
2 operations in the intermediate function comprise ~~include~~ a pause for receiving user input before
3 or after the direct call to the executable function.

1 10. (Previously Amended) A computer program product for use on a computer wherein
2 queries are entered by a user for retrieving data in a relational database management system
3 having a query optimizer for generating an access plan for executing the query, said query

optimizer including a direct call mechanism replacing the lookup function of a run-time

interpreter, said computer program product comprising:

a recording medium;

means recorded on said recording medium for instructing said computer to perform the steps of:

(a) determining an executable function associated with a first operation code in the access plan, the first operation code being one of a plurality of operation codes; and

(b) augmenting said first operation code in the access plan with a pointer to said executable function to provide a direct call mechanism replacing a lookup function of a run-time interpreter.

11. **(Original)** The computer program product as claimed in claim 10, the means for instructing said computer further comprising repeating steps (a) and (b) for the remaining operation codes in the access plan.

12. **(Currently Amended)** The computer program product as claimed in claim 10, wherein said step (b) comprises:

augmenting said first operation code in the access plan with a replacement pointer to an intermediate function, ~~said intermediate function including~~ having a data structure, and ~~for storing a~~ the pointer to said executable function in the data structure.

13. **(Currently Amended)** The computer program product as claimed in claim 12, wherein said data structure comprises ~~includes~~ means for storing information associated with said executable function or said first operation code.

1 14. **(Original)** The computer program product as claimed in claim 10, wherein said step (b)
2 comprises augmenting said first operation code in the access plan with another pointer to a data
3 structure, said data structure providing means for storing information associated with said first
4 operation code or said executable function.

1 15. **(Currently Amended)** The computer program product as claimed in claim 10, wherein
2 ~~said step (a) further comprising a step of~~ includes assessing the executable function associated
3 with the first operation code to determine whether a more specific executable function exists and,
4 if found applicable, replacing the call to the executable function with the a call to a second more
5 specific executable function.

1 16. **(Currently Amended)** The computer program product as claimed in claim 12, wherein
2 said intermediate function comprises ~~includes~~ processing operations for the first operation code
3 or the executable function associated with the first operation code.

1 17. **(Currently Amended)** The computer program product as claimed in claim 16, wherein
2 said processing operations in the intermediate function comprise ~~include~~ gathering statistics on
3 the use of the executable function associated with the first operation code.

1 18. **(Currently Amended)** The computer program product as claimed in claim ~~16~~ 12,
2 wherein said processing operations in the intermediate function comprise ~~include~~ a pause for
3 receiving user input before or after a the direct call to the executable function.

1 19. **(Previously Amended)** A relational database management system for use with a
2 computer system wherein queries are entered by a user for retrieving data from tables, the

3 relational database management system including a query optimizer for generating an access
4 plan associated with the queries entered by the user, said query optimizer including a direct call
5 mechanism replacing a lookup function of a run-time interpreter, said relational database
6 management system comprising:

7 (a) means for determining an executable function associated with each of a plurality
8 of operation codes in the access plan; and

9 (b) means for augmenting said operation codes in the access plan with a pointer to
10 said executable function associated with each operation code to provide a direct call mechanism
11 replacing a lookup function of a run-time interpreter.

1 20. **(Currently Amended)** The relational database management system as claimed in claim
2 19, wherein said means for augmenting said operation codes further comprises ~~includes~~ means
3 for replacing each said operation codes in the access plan with a replacement pointer to an
4 intermediate function, and wherein said intermediate function comprises ~~including~~ a data
5 structure for storing ~~a~~ the pointer to said executable function.

1 21. **(Currently Amended)** The relational database management system as claimed in claim
2 20, wherein said data structure comprises ~~includes~~ means for storing information associated with
3 said executable function or said operation codes.

1 22. **(Currently Amended)** The relational database management system as claimed in claim
2 19, wherein said means for augmenting said operation codes further comprises ~~includes~~ means
3 for adding another pointer to a data structure, said data structure providing means for storing
4 information associated with said operation codes or said executable function.

REMARKS

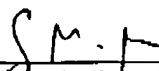
5 Claims 3-9, 12-13, 15-18 and 20-22 are hereby amended. There are still 22 claims in this application.

10 Reconsideration is respectfully requested because the Applicant opines that all claims are in good condition for allowance. None of the cited references discloses the subject matter and features of claims 1-22 of the present invention and, even if they did show some individual features, they would not be able to meet the claims of the present invention which provide new and unexpected results over these references and are thus not anticipated under Section 102 and unobvious and patentable under Section 103.

15 In view of the above, it is submitted that this application is now in good order for allowance, which applicant respectfully solicits. Should matters remain which the Examiner believes could be resolved in a telephone interview, the Examiner is kindly requested to telephone the applicant's undersigned attorney. No additional fee is required in connection with this communication since the number of claims is not extending the original number of claims.
20 However, any underpayment is authorized to be charged to Deposit Account Number 09-0460 in the name of IBM Corporation.

Respectfully submitted,

25 Date: July 23, 2006


Sandra M. Parker
Reg. No. 36,233

30 LAW OFFICE OF SANDRA M. PARKER
329 La Jolla Avenue
Long Beach, CA 90803
Phone/Fax: (562) 597-7504